UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/795,923	03/08/2004	Daniel Kegel	I004-P03074US	1818

33356 7590 10/24/2008

SoCAL IP LAW GROUP LLP
310 N. WESTLAKE BLVD. STE 120
WESTLAKE VILLAGE, CA 91362

EXAMINER	
JEAN GILLES, JUDE	

ART UNIT	PAPER NUMBER
2443	

MAIL DATE	DELIVERY MODE
10/24/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/795,923
Filing Date: March 08, 2004
Appellant(s): KEGEL, DANIEL

Mark A. Goldstein
Reg. No. 50,759
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 08/06/2008 appealing from the Office action mailed on 03/12/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,091,802	Smith et al	07-2000
2005/0257109	Averbuj et al	11-2005

2003/0067874	See et al	04-2003
2003/0107596	Jameson, Kevin Wade	06-2003

(9) *Claim Rejections - 35 USC § 103*

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. **Claims 1-2, 5-12, 15-23, 26-28, and 31-33** are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith et al (hereinafter Smith), U.S. Patent No. 6,091,802 in view of Averbuj et al (hereinafter Averbuj), U.S. Pub. No. 2005/0257109 A1.

Regarding claim 1, Smith teaches the invention substantially as claimed. Smith discloses a method of creating network traffic replicating activities of a large number of users in a telecommunications system (*see abstract*) comprising:

receiving a test script including a plurality of commands (*column 3, lines 40-48, continue in lines 59-67*)

invoking a script interpreter (*fig. 1, tester 100; column 3, lines 25-29, and column 4, lines 14-18; tester 100 is the script interpreter when invoked is capable of running and interpreting command and data of the test scripts*)

launching an application thread to execute the test script (*column 4, lines 61-65).*

However, Smith does not specifically disclose "invoking a protocol engine for each of the commands in the test script such that each protocol engine has an associated command", and "each protocol engine executing its associated commands". In the same field of endeavor, Averbuj shows a mechanism for testing a telecommunication system by associating each Protocol engine to a testing algorithm command. Averbuj discloses "*...In particular, algorithm controller 26 sequentially delivers each command of the selected algorithm to sequencers 8, and proceeds from one command to the next upon receiving an acknowledge signal from each of sequencers 8. In this manner, algorithm controller 26 ensures that each sequencer 8 has completed application of a current command to memory modules 12 via memory interfaces 10 before proceeding to the next command...*" (see Averbuj, par. 0035, 0041, and 0042). The protocol sequencers are protocol engines and that each sequencer interprets the commands from the testing algorithms based on a command protocol (*see Averbuj, abstract*). In an attempt to facilitate simultaneous application of algorithms that contain many protocol commands to different modules applying each sequencer or engine its associated command makes sense in that it reduces the overall test time (see par. 12, and 0016).

Given this feature, a person of ordinary skill in the art would have readily recognized the desirability and advantages of modifying the system shown Smith, to employ the features disclosed by *Averbuj* in order to offer the flexibility of allowing a variety of test algorithms to easily be defined and maintained centrally in the form of

generalized commands, thereby eliminating the need to store common test algorithms in a distributed fashion (see *Averbuj, par. 0015).* By this rationale, claim 1 is rejected.

Regarding claims 2, 5-12, 15-23, 26-28, and 31-33 the combination Smith-Averbuj discloses:

2. The method of claim 1 wherein the commands in the test script simulate actions taken by a network user (*see Smith, column 2, lines 19-32; column 3, lines 30-38*).

5. The method of claim 1 wherein the test script causes network traffic to be produced (*see Smith, abstract; also see column 4, lines 58-64; note "the execution of test scripts for transmitting voice and digital data, detecting voice and digital data, and evaluating voice and digital data…"*).

6. The method of claim 1 wherein each protocol engine executing its associated command comprises:

checking whether a maximum number of protocol engines has been exceeded performing the executing when the maximum number of protocol engines has not been exceeded. (*see Averbuj; par 0035, Smith teaches "In this manner, algorithm controller 26 ensures that each sequencer 8 has completed application of a current command to memory modules 12 via memory interfaces 10 before proceeding to the next command. Algorithm controller 26 may be programmatically or statically configured to establish the number of device blocks 6 and, in particular, sequencers 8 that are present within*

electronic device 2..." Because each controller (protocol engine) has to complete application of a current associated command, and that the number of available sequencers is taken into consideration, one can conclude that if the maximum of protocol engines has been exceeded, the next command had to wait and cannot be immediately executed

7. The method of claim 6 wherein the checking further comprises: waiting for a system defined amount of time until attempting to execute again (*see Averbuj; fig. 3; par. 0035;* 0037-0038; the amount of waiting time here is proportional to the time programmatically required by controller 26 to ensure that each sequencer 8 has finish completing its current command operation).

8. The method of claim 6 wherein the checking further comprises: sleeping until system resources sufficient for the executing of the protocol engine are available until attempting to execute again (*see Averbuj; fig. 3; par. 0035;* 0037-0038)

9. The method of claim 1 wherein the network traffic is comprised of a plurality of data units adhering to a plurality of communications protocols (*see Smith, column 5, lines 48-61*).

10. The method of claim 9 wherein the plurality of communication protocols includes at least one of Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol

(TCP), Internet Protocol (IP), File Transfer Protocol (FTP), or Hypertext Transfer Protocol (HTTP) (*see Smith, column 10, lines 59-65*).

Claim 11 is similar in scope to claim 1, but is recited in the form of a machine readable medium instead of a method. Claim 11 is rejected for the same reasons specified for the rejection of claim 1 above.

Claim 12 is similar in scope to claim 2, and is rejected for the same reasons specified for the rejection of claim 2 above.

Claim 15 is similar in scope to claim 5, and is rejected for the same reasons specified for the rejection of claim 5 above.

Claim 16 is similar in scope to claim 6, and is rejected for the same reasons specified for the rejection of claim 6 above.

Claim 17 is similar in scope to claim 7, and is rejected for the same reasons specified for the rejection of claim 7 above.

18. The machine readable medium of claim 11 coupled with a network testing system (*see Smith, column 3, lines 10-18; fig. 1; see also Smith, column 1, lines 13-17*).

19. The machine readable medium of claim 18 wherein the network testing system is

coupled to a production network (*see Smith, column 3, lines 10-18; fig. 1, switched telephone network 104 is the production environment network coupled to the system tester 100; see also Smith, column 1, lines 13-17*).

20. The machine readable medium of claim 19 wherein the network testing system is coupled to a test network (*see Smith, column 3, lines 10-18; fig. 1, communications system 102 network is a test network connected to system tester 100; see also Smith, column 1, lines 13-17*).

21. A system to create network traffic simulating activities of a large number of users (see Smith, abstract), the system comprising:

a plurality of script interpreter units in user space (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65; processors 222 each is a script interpreter unit scheduling and controlling the execution of the test scripts running on user test computer 202*)

each script interpreter unit to interpret a script including a plurality of commands (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65*),

an application thread in user space for each script interpreter unit (*see Smith;, column 4, lines 58-65*)

a plurality of protocol engines in user space for each application thread, each protocol engine to execute a command included in one of the scripts is needed for the specified protocol, to execute the matching test script command (*see Averbuj, par.*

0035, 0041, and 0042; the sequencers 8 which are the protocol engines operating device 2 and executing a test script command as specified by the associated protocol),

an operating system in operating system space (*see Smith, column 4, lines 44-50;an example of the operating system in question here is WINDOWS NT*).

The same motivation and reason to combine used in the rejection of claim 1 is also valid for this claim. By this rationale, claim 21 is rejected.

22. The system of claim 21 wherein the system supports a plurality of communications protocols (*see Smith, column 5, lines 48-61*).

23. The system of claim 22 wherein the plurality of communications protocols includes at least Ethernet, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), Internet Protocol (IP), and Hypertext Transfer Protocol (HTTP) (*see Smith, column 10, lines 59-65; see also Smith column 1, lines 1-17 whereas it is disclosed " …Telecommunication networks, such as conventional public or private switched telephone networks and more recently packet-switched networks and the Internet, interconnect human operators and telecommunication systems, which commonly run telephony applications including voice-mail, telephone banking systems, automated directory assistance, and multi-branched telephone customer service systems…". A person of ordinary skill in the art knows that such telecommunication systems are capable of supporting and running on a plurality of protocols such as UDP, TCP, IP, and HTTP*).

26. A system to create network traffic simulating the activities of a large number of users (*see Smith; abstract*), the system comprising:

a plurality of script interpreter units in user space (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65; processors 222 each is a script interpreter unit scheduling and controlling the execution of the test scripts running on user test computer 202*),

each script interpreter unit to interpret a script including a plurality of commands (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65*),

an application thread in user space for each script interpreter unit (*see Smith;, column 4, lines 58-65*),

a plurality of protocol engines in user operating system space for each application thread, each protocol engine to execute a command included in one of the scripts (*see Averbuj, par. 0035, 0041, and 0042; the sequencers 8 which are the protocol engines operating device 2 and executing a test script command as specified by the associated protocol*).,

an operating system in operating system space (*see Smith, column 4, lines 44-50;an example of the operating system in question here is WINDOWS NT*). The same motivation and reason to combine used in the rejection of claim 1 are also valid for this claim. By this rationale, claim 26 is rejected.

Claim 27, and 32 are similar in scope to claim 22, and are rejected for the same reasons specified for the rejection of claim 22 above.

Claim, 28, and 33 are similar in scope to claim 23, and are rejected for the same reasons specified for the rejection of claim 23 above.

31. A system to create network traffic simulating activities of a large number of users (see Smith, abstract), the system comprising:

a plurality of script interpreter units in user space (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65; processors 222 each is a script interpreter unit scheduling and controlling the execution of the test scripts running on user test computer 202*)

each script interpreter unit to interpret a script including a plurality of commands (*see Smith; abstract, column 3, lines 49-67, column 4, lines 44-65*),

an application thread in operating system space for each script interpreter unit (*see Smith;, column 4, lines 58-65*),

a plurality of protocol engines in user operating system space for each application thread, each protocol engine to execute a command included in one of the scripts (*see Averbuj, par. 0035, 0041, and 0042; the sequencers 8 which are the protocol engines operating device 2 and executing a test script command as specified by the associated protocol*)

an operating system in operating system space (*see Smith, column 4, lines 44-50;an example of the operating system in question here is WINDOWS NT*). The same motivation and reason to combine used in the rejection of claim 1 are also valid for this claim. By this rationale, claim 31 is rejected.

5. **Claims 3-4, 13-14, 24-25, 29-30, and 34-35** are rejected under 35 U.S.C. 103(a) as being unpatentable over Smith and Averbuj, in further view of Jameson U.S. Pub. No. 20030107596 A1

Regarding claim 3, the combination of Smith-Averbuj teaches the invention in substance as claimed. Smith-Averbuj discloses the method of creating network traffic as per claim 1, but fail to disclose the details of a method wherein the commands in the test script include extended operation operating system commands.

In an analogous art, Jameson shows the use of work operations, specific computer programs, or computer scripts that carry out computer actions through the use of commands. Jameson discloses "*The default set of work operations made available by a computer operating system is called the default command set of the operating system. In practice, default operating system command sets are always extended with additional programs to provide users with application-specific work operations or commands. Thus the total set of work operations available in a typical command line shell window is the union of the default operating system command set and the*

additional application-specific work operation set" (*See Jameson par. 0018*). In order to increase the total number of available work operations or commands in a typical software environment this approach works, as it can increase the command list options for effectively testing the system.

Accordingly, it would have been obvious for an ordinary skill in the art, at the time the invention was made to have incorporated the invention of Jameson with the teaching of Smith and Averbuj, for the purpose of " automating systems to replace manual human effort thereby improving the productivity of software developers, web media developers, and other humans that work with collections of computer files" (*see Jameson, par. 0006-0008*). By this rationale, **claim 3** is rejected.

Regarding claims 4, 13-14, 24-25, 29-30, and 34-35, the combination Smith-Averbuj-Jameson discloses:

4. The method of claim 3 wherein the extended operation operating system commands include "fetch," "verify," "fetch and verify," "fetch and ignore," "monitor," and "count" (*see Jameson par. 0018*). While Jameson does not spell out in his teaching the specific commands indicated above, Jameson teaches default operating system command sets that are extended with additional programs to provide users with application-specific commands. Accordingly it would have been obvious for a person of ordinary skill in the art to have incorporated specific commands such as "fetch," "verify," "fetch and verify," "fetch and ignore," "monitor," and "count" in the command sets of Jameson for the purpose of automating the systems and improving the productivity as stated by

Jameson in par. 0006-0008. The same motivation and reason to combine Jameson with Smith and Averbuj, used in the rejection of claim 1 is also valid for this claim. By this rationale, claim 4 is rejected.

Claims 13, 24, 29, and 34 are similar in scope to claim 3, and are rejected for the same reasons specified for the rejection of claim 3 above.

Claim 14, 25, 30, and 35 are similar in scope to claim 4, and are rejected for the same reasons specified for the rejection of claim 4 above.

(9) Response to Arguments

Point A) The examiner rejected claims 1-2, 5-12, 15-23, 26-28 and 31-33 were rejected as obvious in view of Smith and Averbuj. Claims 3-4, 13-14, 24-25, 29-30 and 34-35 were rejected as obvious in view of Smith, Averbuj and Jameson. We assert that these rejections are not proper because the cited references may not properly be combined.

Appellants contend that the references are so unrelated that it would not be logical to combine different aspects to result in the claimed subject matter, for the following reasons:

Issue 1) Succinctly, Smith discloses a telecommunications testing system and Averbuj discloses a memory testing system. These publications disclose different techniques that solve problems that are wholly unrelated to one another. Stated another way, Smith teaches a computer that performs telecommunications testing over a

telecommunications network. Averbuj teaches a controller that performs memory testing of a single electronic device. The low level, chip based testing techniques of Averbuj exist in a different realm and are not applicable to the higher level, inter-device and telecommunications network testing techniques taught in Smith. Simply, Smith and Averbuj are in different technology spaces. As such, their teachings may not be properly combined. And that Averbuj is simply restricted to testing memory modules in a single electronic stand alone device.

response to issue 1) The Examiner disagrees with the Appellants' remarks. Smith teaches "An apparatus and method for testing a telecommunication system, which runs telephony applications such as voice-mail, telephone banking systems, automated directory assistance, and multi-branched telephone customer service systems. The apparatus includes a test computer for scheduling and controlling the execution of test scripts" see Smith Abstract. Averbuj teaches a controller to perform memory testing as stated in the abstract. However, these memory modules as specified by Averbuj are designed in such a way that the memory modules can be integrated into a telecommunication network. In par. 0032, Averbuj teaches "electronic device 2 may be any device that incorporates memory modules, such as an embedded computing system, a computer, server, personal digital assistant (PDA), mobile computing device, mobile communication device, digital recording device, network appliance, mobile positioning device, and the like...". Clearly the testing techniques of Averbuj are not retricted a stand alone device, but in the context of a communication network network.

Issue 2) Appellants argue further that Smith and Averbuj are in wholly different patent classes, and, as such, may not be properly combined. Smith is in class 379/29 and various other subclasses in class 379. Class 379 is for "telephonic communications" while subclass 29.01 is more specifically directed to "terminal arrangement to enable remote testing (e.g., testing interface)". Differently, Averbuj was published in class 714/33 which is directed to "error detection/correction and fault detection/recovery derived from analysis (e.g., of a specification or by stimulation)". Now in prosecution, Averbuj has been placed in class 714/718. Class 714/718 is specifically directed to "memory testing". That the U.S.P.T.O.'s classification system shows that Smith and Averbuj are in wholly different patent classes supports the conclusion that the references may not be properly combined.

response to issue 2) The Examiner disagrees with this point of contention regarding the combination of Smith and Averbuj. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Furthermore,

Issue 3) Appellants argue that Smith discloses that it uses test scripts that include commands involving telephony functions. (See Smith, 5:56-61) There is no problem in Smith's use of test scripts; the scripts achieve their intended purpose. In the *KSR* case, the Supreme Court qualified the issue of hindsight by stating that rigid preventative rules

that deny factfinders recourse to common sense, however, are neither necessary under our case law nor consistent with it." *KSR Int'l Co. v. Teleflex, Inc.,* 127 S.Ct. 1727 (2007). In the instant matter, a person of ordinary skill in the art having common sense at the time of the invention would not have reasonably looked to Averbuj to solve a problem with scripts or test algorithms, because Smith did not have any such problems to be solved. See *Exparte Rinkevich et al.,* Appeal 20061317 (BPAI May 29, 2007). It can only be from improper hindsight that the Examiner is attempting to combine Smith and Averbuj. Since Smith already teaches a solution, it is clear that the Examiner has used the claims as a guide or roadmap in formulating the rejection. As such, the references may not be properly combined to make an obviousness rejection.

Response to issue 3) In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). See rejection of claim 1, and Issue 1 response above.

Point B) All Claims: The References Do Not Disclose The Claimed "Protocol Engine". The independent claims, claims 1, 11, 21, 26 and 31, recite a "protocol engine". The Examiner directs us to Averbuj for a teaching of protocol engines. The

Examiner asserts that the sequencers and test algorithms of Averbuj teach the claimed protocol engine. However, the sequencers and test algorithms of Averbuj fail to teach or suggest the protocol engine claimed.

Answer to Point B) The Examiner disagrees with appellants argument. The Sequencers as defined are protocol engines that conform to a command protocol to test a plurality of application modules (see par. 0014, 0029, 0046).

Point C) Claims 2 and 12: The Combination of References Can Not Be Cited For Teaching Simulating Actions Taken by a Network User. Claims 2 and 12 recite that "the commands in the test script simulate actions taken by a network user". We assert that Smith cannot be combined with Averbuj to teach "the commands in the test script simulate actions taken by a network user".

Answer to point C) Again the Examiner refers the appellants to the rejection of claims 2 and 12 above.

The same arguments used in the rejections of claims 1, 11, 21, 26, and 31 are also valid for points D, F, G, H, and pertinent claims as rejected above.

For the above reasons, it is believed that the rejections should be sustained.

Conclusion

Any inquiry concerning this communication or earlier communications from examiner should be directed to Jude Jean-Gilles whose telephone number is (571) 272-3914. The examiner can normally be reached on Monday-Thursday and every other Friday from 8:00 AM to 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Nathan Flynn, can be reached on (571) 272-1915. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3301.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (571) 272-0800.

Respectfully submitted,

/Jude J Jean-Gilles/

Primary Examiner, Art Unit 2443

October 20, 2008